Using Google Apps™ Credentials for WiFi Authentication

WPA2/802.1X and Captive Portal based Authentication

Cloudessa, Inc.
Palo Alto, CA
July 2013

Overview

There are now over 5 million schools, businesses, government agencies and other institutions that have established Google Apps* domains, with tens of millions of users, now using Google® Apps.

In connection with the creation of an organization's Google Apps domain, domain owners specify how users are to be authenticated to that domain, typically based on username and password.

Users are then invited and required to create a user name and password for secure access to that particular Google Apps domain, and assigned to a group for application layer access within that domain.

With Cloudessa RADIUS, organizations can now re-use these Google Apps user name and password account credentials, and group assignment for securing user access to the WiFi network.

Cloudessa RADIUS:

- eliminates the need for domain owners to set-up and maintain a separate database of user names and password for WiFi network access
- can save administrative time, money, and hassle by enabling the re-use of existing Google Apps account credentials for WPA 2 / 802.1X or Captive Portal based WiFi access security
- runs in the Cloud, removing the need for, and expense of, having on-premises RADIUS servers. You get the cost and management efficiencies of a cloud service infrastructure, while your critical Google Apps user account credentials remain fully under your control.

Authentication Options: WPA 2 / 802.1X or Captive Portal

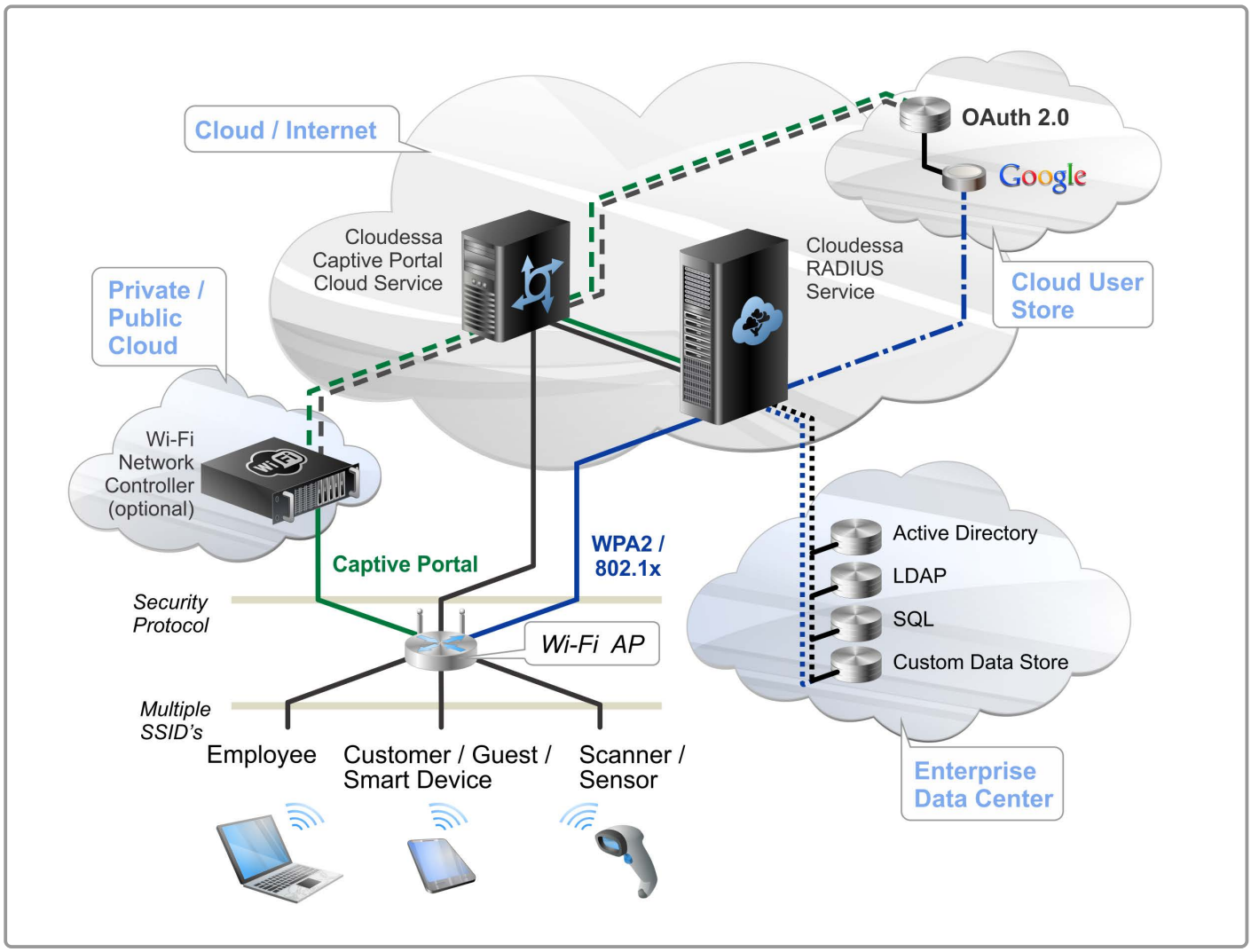
Organizations can authenticate users to their WiFi network using Google Apps using either WPA 2 / 802.1X or Captive Portal based access security.
Best practice for WiFi access to enterprise LAN applications mandates the use of WiFi Protected Access 2 Enterprise (WPA2) and 802.1X-based security; in addition, WPA2 and 802.1X are considered essential for securing WiFi access in healthcare (HIPAA), financial services (SOX), and other regulated environments.

If the primary use of the WiFi network is to access cloud or external resources, (for instance in a hotspot or for student / customer / guest internet access) or if a users session will be protected via a VPN tunnel and there is little risk of sensitive data being compromised, then Captive Portal is a viable option.

* *Google Apps is a trademark of Google, Inc.*

- **WPA 2 / 802.1X authentication (EAP-TTLS/PAP option)**
  User enter their Google user name and password using EAP-TTLS / PAP protocol. This protocol is supported out of the box on Android, OS X, iOS and Windows 8 platforms. On older Windows platforms the user needs to install an EAP-TTLS client, such as SecureW2.
- **WPA 2 / 802.1X authentication (EAP-TLS option)**
  Cloudessa provides a functionality to create a Digital Certificate for your Google Apps users. The certificate is passed to the RADIUS Server during the EAP-TLS based authentication process, validated, and used to authenticate the user to the network. EAP-TLS is supported on all Android, Windows, iOS and OS X platforms and does not require installation of any additional client software.
- **Captive Portal based authentication.**
  With a captive portal, a user accessing the WiFi network is redirected to the portal page, and prompted to enter Google authentication credentials. This method does not require installation of any additional client software.

Cloudessa RADIUS supports both WPA 2 / 802.1X and Captive Portal Browser based access security.



Cloudessa RADIUS supports both WPA2 / 802.1X and Captive Portal (via UAM compliant WiFi AP or optionally via a UAM complaint WiFi Controller) based authentication using Google Apps, Active Directory or LDAP user stores.

Layer 2 vs. Layer 3 Authentication

WPA 2 /802.1X works at Layer 2, the data link layer. In this case, the wireless client is authenticated, the encryption key is derived and the Layer 2 wireless connection between the client and the access point is encrypted.

Captive Portal authenticates users at Layer 3, the network layer. In this case the encryption is typically done at the level of the browser using the HTTPS protocol. Alternatively, a layer 3 VPN (such as IPSec or SSL VPN) can be used to encrypt the entire layer 3 traffic).

When assessing your WiFi network security requirements, it is important to examine what is the right level of security for your deployment, and how do you want to enforce the access security.

With 802.1X, authentication happens before a user is granted an IP address and allowed on the network, this protects against attacks at upper layers by denying access before a rogue user ever gets on the network. In addition, RADIUS attributes can be used to set user parameters, such as VLAN, Quality of Service, and group policy.

Captive Portal browser-based login is an application-level authentication. With Captive Portal, the user does obtain an IP address on the network prior to authentication; however, their network usage is restricted until they are authenticated via a browser-based login.

Cloudessa RADIUS and Google Apps credentials can be used to authenticate users at either level, depending on your current infrastructure and security risk profile. For example, organizations who's employees will be using the WLAN to access corporate applications and resources and cannot risk their network or data being compromised should consider the more secure Layer 2 security approach.

If the primary use of the WiFi network is to access cloud or external resources, (for instance in a hotspot or for customer / guest access) or if a users session will be protected over the WLAN via a VPN connection, and there is little risk of sensitive data being compromised, then Layer 3 Captive Portal security is an appropriate option.

The Role of RADIUS and AAA

Regardless of which method you choose for enforcing access security on your WiFi network, authenticating users to a network through client based WPA2 / 802.1X or Captive Portal, both require the use of a RADIUS server.

The RADIUS server provides the means to centrally manage authentication, authorization, and accounting (AAA).

A combination of different network elements work collaboratively to manage and secure WiFi deployments.

- A centralized RADIUS server accepts authentication requests from WiFi access points and controllers. User credentials are then processed against a designated user store, in this case, Google Apps.
- Authentication is accepted or rejected based on the validity of the provided user account credentials.
- Once the user is authenticated, the users Authorization to network resources (what they can do while on the network) is based on attributes returned by the RADIUS server for each user session, based on which group or groups the user is an authenticated member of (based on the users group assignments in Google Apps or other user store).

The role of the RADIUS server is essential, not only does it authenticates the user, but it also communicates back to the WiFi AP or controller (via RADIUS attributes), the parameters for how the AP should be configured for that particular user, for that particular session, based on what network group (as defined in Google Apps or other user store) that the user is a member of. Such parameters can include assigning users to particular VLAN's, setting bandwidth allocation, and dynamically configuring any other configurable policy element of your access gateway.

- RADIUS accounting logs detailing user and device access are generated and stored.

WiFi access security is dependent on the interoperability between a number of different network components, including:

- User Device, typically a laptop or smart device running "client" or "supplicant" software or a browser;
- WiFi AP, WiFi Controller - The AP is the access security enforcement point and is the "Authenticator" or "RADIUS Client" that initiates and sends the RADIUS authentication request to the RADIUS server;
- RADIUS Server - Standards based server that handles the authentication, authorization, and accounting for user access;
- User Store - Google Apps, Active Directory, or other user store where user credentials and user group assignments are stored.

All of these network components must be configured and interoperable to enforce WiFi security.

WPA 2 / 802.1X Based Access Security - Details

EAP-TTLS Authentication

To authenticate users to the WiFi network using Google Apps domain account user names and passwords requires using EAP-TTLS / PAP (Extensible Authentication Protocol with Tunneled Transport Layer Security / Password Authentication Protocol). for securely passing the users account credentials from the user device to the network and over to Google for authentication and authorization.

To authenticate a user using their Google Apps user name and password, EAP-TTLS must be the outer authentication, while PAP must be used as the inner authentication protocol.

The authentication flow proceeds as follows:

- EAP-TTLS / PAP first authenticates the connection between the WiFi AP (the "Authenticator" or RADIUS Client) and the RADIUS server and sets up a trusted secure EAP-TTLS tunnel between the Authenticator and the RADIUS server.
- Once the EAP-TTLS tunnel is established between the client and the RADIUS server, the client will send authentication credentials as PAP protocol messages within the encrypted EAP-TTLS tunnel. The credentials pass through the AP in encrypted form. The RADIUS server will then receive the credentials and pass them to Google Apps cloud service for verification.

While many mobile device and latest OS support EAP-TTLS/PAP natively one might require installing an 802.1X supplicant for older devices to support EAP-TTLS / PAP.

802.1X Supplicants

The following Operating Systems all include 802.1X supplicants and support EAP-TTLS and PAP:

- Apple, iOS version 3.1.3 and higher and MAC OS ;
- Android v2.1 and higher;
- Google Chrome™ OS (for Chromebooks);
- Microsoft Windows v8+ (note: Windows Mobile does note support EAP-TTLS;
- Blackberry 6A+.

Administrators can automate user supplicant configuration through the use of profile creation tools (ie: iOS Profiles) and scripting. Alternatively, SecureW2's "JoinNow MultiOS" is a wireless security deployment platform that includes a client with support for a full range of Extensible Authentication protocols (EAP) including EAP-TTLS/PAP.
See *www.securew2.com*.

Please visit *www.cloudessa.com/support* for detailed information about configuring the various supplicants for EAP-TTLS / PAP, profiling and scripting tips, and the latest information about other operating systems.

Note that in addition to WiFi, EAP-TTLS/PAP is supported by many wired Layer 2 switches, Layer 2/3 gateways, and VPN devices. Therefore, Cloudessa can also be used to authenticate access to the wired network using Google Apps.

EAP-TLS authentication

In lieu of user names and passwords, Google Apps domain owners can opt to issue X509 certificates to their Google Apps users and use them with EAP-TLS protocol for user authentication.

EAP-Transport Layer Security (TLS) is used in certificate-based security environments, providing mutual authentication, negotiation of the encryption method, and encrypted key determination between the client and the authenticating server.

To enable the use of certificate credentials in a WPA 2 compliant manner, a signed certificate must first be in the certificate store on the mobile device, and then the user must present that certificate during the WiFi authentication process using a EAP-TLS supplicant.

Cloudessa Certificate Creation Tool

To facilitate the creation and distribution of Certificates signed by Google Apps, Cloudessa has created a Certificate Creation Utility, that administrators can use to easily create certificates on behalf of their Google Apps users. The tool enables the importing of user names and email addresses, the generation of signed certificates, and it automates the process of then sending the certs to user via email for easy insertion into the certificate store on their device(s).

The authentication flow proceeds as follows:

- The client and the RADIUS server establish a connection through the AP.
- The client sends its digital certificate to the server, the server sends its digital certificate to the client.
- The client verifies that a trusted third-party authority signs the server certificate, the server verifies that the client certificate is signed by Cloudessa.

- The client and the server prove to each other the possession of the private keys for each certificate by signing random challenges.
- The server verifies that the client certificate is issued for the correct Google Apps domain by matching the domain name in the certificate. The server also verifies that the client certificate has not been revoked.

The admin will use Cloudessa to:

- create and sign certificates for each user
- email certificate-installation links to users.

The users install the certificates by simply clicking the link inside the email. During the EAP-TLS based authentication, the certificate is validated, and the email address of the certificate owner is checked against a listing of current Google Apps domain users maintained in the Cloudessa native database which is regularly synched with Google Apps.

When a user is deleted in Google Apps, the user certificate is revoked.

In case when a mobile device is lost Cloudessa provides an interface to revoke the certificate installed on the lost device and generate a new certificate for the user.

Note that in addition to WiFi, EAP-TLS is supported by many wired Layer 2 switches, Layer 2/3 gateways, and VPN devices. Therefore, Cloudessa can also be used to authenticate access to the wired network using Google Apps.

Captive Portal - Details

To allow user access to your WiFi network, you can configure the Cloudessa Captive Portal Cloud Service to provide a Web-based login mechanism for users to enter their login credentials.

Cloudessa Captive Portal is based on the Universal Access Method (UAM) protocol, which is supported by all major enterprise access point manufacturers.

Cloudessa's support for the exchange of Google Apps Credentials via a Captive Portal browser based login is based on OAuth (Open Standard for Authorization). OAuth provides a standard method for end-users to authorize third-parties like Cloudessa to access Google on their behalf without sharing their credentials, using user-agent redirections.

The authentication flow proceeds as follows:

- User associates to the WiFi network, the WiFi access point, using UAM, redirects the user browser to the Cloudessa Captive Portal service.
- Upon redirect, the user lands on a domain specific Captive Portal web page. On this page, is a login prompt for Google Apps. The login window is actually a secure redirect back to the Google login page, so the credential exchange is with Google itself, not with Cloudessa or the Portal.
- If the authentication credentials the user presents are valid, Google will communicate a success message to the Captive Portal server
- The Captive Portal server creates temporary RADIUS credentials on the RADIUS server and sends a message to the AP which includes temporary RADIUS credentials.
- The AP then initiates a RADIUS authentication request using these credentials.
- The Cloudessa RADIUS server responds to the request with a set of RADIUS attributes that controls user session parameters, such as VLAN id and Quality of Service.

Another benefit of the Cloudessa Captive Portal is that when a user logs in using their Google Apps credentials, Google stores a single-sign-on cookie in their browser. This then enables them to seamlessly access Google Apps resources such as Gmail™, Google Docs™ and Google Drive™ without the need to further authenticate to Google, saving time and improving user experience.

Using Active Directory with Google Apps

Many organizations want to keep their directory services in-house, and continue to use Active Directory (AD) or an LDAP or SQL database as a user store even after they have migrated to Google Apps.

Cloudessa RADIUS can be easily configured to support authenticating users directly against Active Directory or an LDAP database. In this use case, a user would present their credentials to the WiFi AP via their 802.1X supplicant. The supplicant would then pass the users credential to the Cloudessa RADIUS server; the Cloudessa RADIUS server would then authenticate the user against the appropriate AD or LDAP user store.

Cloudessa, Inc.
2225 East Bayshore Road, Suite 200
Palo Alto, CA, 94303

Email Us:
sales@cloudessa.com
support@cloudessa.com